Mbrane

Logic Programming

Group Mbrane_4

João Pedro Pinheiro de Lacerda Campos (up201704982) up201704982@fe.up.pt

Leonardo Fernandes Moura (up201706907) up201706907@fe.up.pt

15.10.2019

Table of Contents

Game presentation

Origin

Mbrane is a fairly recent game. It was created in 2013 by someone who goes by the name of DukeZhou. When he was a kid, DukeZhou really liked to play strategy board games, but he didn't like games where luck was a relevant factor, he liked games where the player with superior strategy or the one "who played better" was definitly the winner. Games like chess or checkers were all he was into. In 2005 he discovered Sudoku, and even though puzzles were not his favorite type of game, he was amazed by the game. The sheer amount of combinations for a 9x9 board is around 6.7*10^21. That factor combined with the all restraints the puzzle imposes made it feel somewhat magical.
In May 2013 when he was playing Fallout 3 the rules of Mbrane dropped into his head all at once. He immediately called a friend whom he used to play chess with and they spent hours playing and discovering the game, getting more and more fascinated with it as time went by.

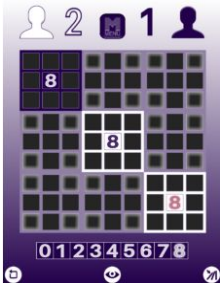
The full story, taken from the journals of DukeZhou can be read in the following address: http://mbranegame.com/origin-of-mbrane.

Rules

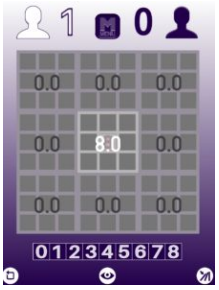
The following rules are the official rules for Mbrane, in their integral redaction.
Mbrane is divided in two distinct phases: the placement phase and the resolution phase

Placement phase

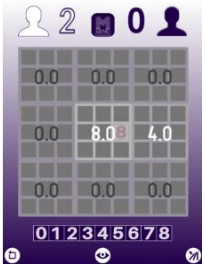
- Players take turns placing numbers in an empty Sudoku. These are bids for territories.



- Players receive points equal to the number in the region where the number is placed. These points are known as **Power**.



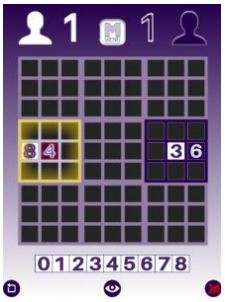
- If a number borders other regions, horizontally, vertically or diagonally, the player receives 1/2 points in those border regions. These points are known as **Influence**.



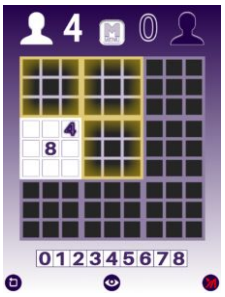
- Placement conforms to the rules of Sudoku, so a number may only be placed once in a region, row or column. (Broken Sudoku will occur.)
- When all numbers that can be placed have been placed, the gameboard is resolved. **Resolution** determines the outcome.

Resolution phase

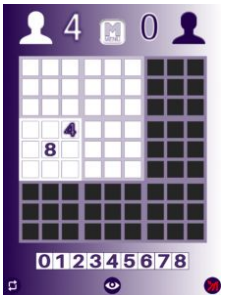
- Regions are resolved in order of the greatest disparity. These are the regions with greatest point difference between players.



- When a region is resolved, it is awarded to the player with the most points in the region, known as the dominant player.
- All opposing numbers in that region defect, flipping to the dominant player. This player is now said to have control.



- If these numbers border other regions, they switch their **Influence** points. (This can shift the balance of power, affecting the outcome of the game!)



- When all regions that can be resolved, have been resolved, the player controlling the most regions wins.
- Strength of victory is measured by the ratio of controlled regions between players, such as 5/4, 4/5, 4/4, 9/0, 0/9, …

The official rules can be consulted in the address: http://mbranegame.com/rules-of-m

Initial Prolog approach

Representation of game states

Although the user can only be able to see one board at a time, in a any phase of the game, there are actually three boards represented in the game's data structures. There's a **board** predicate that represents the game board itself, a list of 9 lists each representing a line of the board. This is the board used to print the state of the game. There's also a **board_blocks** predicate. This predicate is also a list of 9 lists, each one representing a 3x3 block of the board, this is used to facilitate calculations involving power in each block and influence in adjacent blocks caused by the placement of a number in the edge of a block. Finally we have a **board_influence** list that stores the values of power (and spread influence) in each block of the board. When a player adds a number to cell, all the boards are updated accordingly.

```
board([
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' ']] ).

board_blocks([
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' '],
    [' ',' ',' ',' ',' ',' ',' ',' ',' ']] ).

board_influence([0,0,0,0,0,0,0,0,0]).
```

Visualization of the game board

The visualization of the game board is done via the function **displayBoard(Board)**. Because the board is the same in both player's perspectives we don't need to pass the player to the predicate, the board will be displayed in the same way, independently of the player whose turn is happening when the function is called.

The function **displayBoard(Board)** works by printing a line at the time recursively using the **draw_line(Line)** predicate.

```
yes
| ?- board3(Board),display_board(Board)
+---+---+---+---+---+---+---+---+---+
| 6 | 4 | 0 | 1 | 5 | 2 | 3 | 8 | 7 |
+---+---+---+---+---+---+---+---+---+
| 5 |   | 8 | 3 | 0 | 7 | 6 | 1 | 4 |
+---+---+---+---+---+---+---+---+---+
| 3 | 1 | 7 | 6 | 8 | 4 | 5 | 2 |   |
+---+---+---+---+---+---+---+---+---+
| 2 | 3 | 6 | 7 | 1 | 5 | 8 | 4 | 0 |
+---+---+---+---+---+---+---+---+---+
| 4 | 5 |   |   | 6 | 8 | 2 | 3 | 1 |
+---+---+---+---+---+---+---+---+---+
| 1 | 7 |   | 2 | 3 |   |   | 5 | 6 |
+---+---+---+---+---+---+---+---+---+
|   |   | 3 | 8 | 7 | 6 | 1 | 0 | 2 |
+---+---+---+---+---+---+---+---+---+
| 0 | 8 |   | 5 | 2 | 1 | 4 | 6 | 3 |
+---+---+---+---+---+---+---+---+---+
| 7 | 6 | 1 | 0 | 4 | 3 |   | 8 | 5 |
+---+---+---+---+---+---+---+---+---+
```

Sources

- All the rules were taken from: http://mbranegame.com/rules-of-m/
- All the images were taken from: http://mbranegame.com/tutorial/
- The story and origins of the game: http://mbranegame.com/origin-of-mbrane/
- Extra information and official website: http://mbranegame.com/